MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 821 February, 1985

# Direct Passive Navigation

Shahriar Negahdaripour
Berthold K.P. Horn

**Abstract:** In this paper, we show how to recover the motion of an observer relative to a planar surface directly from image brightness derivatives. We do not compute the optical flow as an intermediate step. We derive a set of nine non-linear equations using a least-squares formulation. A simple iterative scheme allows us to find either of two possible solutions of these equations. An initial pass over the relevant image region is used to accumulate a number of moments of the image brightness derivatives. All of the quantities used in the iteration can be efficiently computed from these totals, without the need to refer back to the image. A new, compact notation allows us to show easily that there are at most two planar solutions.

**Key Words:** Passive Navigation, Optical Flow, Structure and Motion, Least Squares, Planar Surface, Non-Linear Equations, Dual Solution, Planar Motion Field Equations.



This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the System Development Foundation.

## 1. Introduction

The problem of determining rigid body motion and surface structure from image data has been the topic of many recent research papers in the area of machine vision. Two types of approaches, discrete and continuous, have been pursued. In the discrete approach, information about a finite number of points is used to reconstruct the motion [4,5,9,10,13–16]. To do this, one has to identify and match feature points in a sequence of images. The minimum number of points required depends on the number of images. In the continuous approach, the optical flow that is the apparent velocity of brightness patterns is used at every image point [1–3,12,17]. In general, the computation of the local flow field involves not only a least-squares formulation that uses a constraint equation between the optical flow and the image brightness derivatives at every image point, but also an additional constraint, such as the smoothness of the flow field [7,8]. The additional constraint is required because information is lost when the 3-D world is projected onto a 2-D image. Locally, the brightness variations in time-varying images only provide one constraint on the components of the optical flow. While the continuous approach requires more computation, it is robust with respect to errors in the optical flow data.

Much of the research work in recovering surface structure and motion has been restricted to using the optical flow information in the image. In this paper, we determine the motion of an observer relative to a planar surface directly from the image brightness derivatives, without having to compute the optical flow as an intermediate step. We restrict ourselves to planar surfaces since only three parameters are needed to specify the surface structure. We will be considering less restricting constraints on the surface structure in a later paper.

## 2. Preliminaries

We first recall some details about perspective projection, the motion field, the brightness change constraint equation, rigid body motion and planar surfaces. This we do using vector notation in order to keep the resulting equations as compact as possible.

### 2.1. Perspective Projection

Let the center of projection be at the origin of a Cartesian coordinate system. Without loss of generality we assume that the effective focal length is unity. The image is formed on the plane $z = 1$, parallel to the $xy$-plane, that is, the optical axis lies along the $z$-axis. Let $\mathbf{R}$ be a point in the scene. Its projection in the image is $\mathbf{r}$, where

$$\mathbf{r} = \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \mathbf{R}.$$

The $z$-component of $\mathbf{r}$ is clearly equal to one, that is $\mathbf{r} \cdot \hat{\mathbf{z}} = 1$.

### 2.2. Motion Field

The *motion field* is the vector field induced in the image plane by the relative motion of the observer with respect to the environment. The *optical flow* is the apparent motion of

brightness patterns. Under favourable circumstances the optical flow is identical to the motion field. The velocity of the image $\mathbf{r}$ of a point $\mathbf{R}$ is given by

$$\frac{d\mathbf{r}}{dt} = \frac{d}{dt}\frac{1}{\mathbf{R}\cdot\hat{\mathbf{z}}}\mathbf{R}.$$

For convenience, we introduce the notation $\mathbf{r}_t$ and $\mathbf{R}_t$ for the time derivatives of $\mathbf{r}$ and $\mathbf{R}$, respectively. We then have

$$\mathbf{r}_t = \frac{1}{\mathbf{R}\cdot\hat{\mathbf{z}}}\mathbf{R}_t - \frac{1}{(\mathbf{R}\cdot\hat{\mathbf{z}})^2}(\mathbf{R}_t\cdot\hat{\mathbf{z}})\mathbf{R},$$

which can also be written in the compact form

$$\mathbf{r}_t = \frac{1}{(\mathbf{R}\cdot\hat{\mathbf{z}})^2}\big(\hat{\mathbf{z}}\times(\mathbf{R}_t\times\mathbf{R})\big),$$

since $\mathbf{a}\times(\mathbf{b}\times\mathbf{c}) = (\mathbf{c}\cdot\mathbf{a})\mathbf{b} - (\mathbf{a}\cdot\mathbf{b})\mathbf{c}$. The vector $\mathbf{r}_t$ lies in the image plane, and so $(\mathbf{r}_t\cdot\hat{\mathbf{z}}) = 0$. Further, $\mathbf{r}_t = \mathbf{0}$, if $\mathbf{R}_t \parallel \mathbf{R}$, as expected.

Finally, noting that $\mathbf{R} = (\mathbf{R}\cdot\hat{\mathbf{z}})\mathbf{r}$, we get

$$\mathbf{r}_t = \frac{1}{\mathbf{R}\cdot\hat{\mathbf{z}}}\big(\hat{\mathbf{z}}\times(\mathbf{R}_t\times\mathbf{r})\big).$$

## 2.3. Rigid Body Motion

In the case of the observer moving relative to a rigid environment with translational velocity $\mathbf{t}$ and rotational velocity $\omega$, we find that the motion of a point in the environment relative to the observer is given by

$$\mathbf{R}_t = -\omega\times\mathbf{R} - \mathbf{t}.$$

Since $\mathbf{R} = (\mathbf{R}\cdot\hat{\mathbf{z}})\mathbf{r}$, we can write this as

$$\mathbf{R}_t = -(\mathbf{R}\cdot\hat{\mathbf{z}})\,\omega\times\mathbf{r} - \mathbf{t}.$$

Substituting for $\mathbf{R}_t$ in the formula derived above for $\mathbf{r}_t$, we obtain

$$\mathbf{r}_t = -\Big(\hat{\mathbf{z}}\times\big(\mathbf{r}\times(\mathbf{r}\times\omega - \frac{1}{\mathbf{R}\cdot\hat{\mathbf{z}}}\mathbf{t})\big)\Big).$$

It is important to remember that there is an inherent ambiguity here, since the same motion field results when distance and the translational velocity are multiplied by an arbitrary constant. This can be seen easily from the above equation since the same image plane velocity is obtained if one multiplies both $\mathbf{R}$ and $\mathbf{t}$ by some constant.

## 2.4. Brightness Change Equation

The brightness of the image of a particular patch of a surface depends on many factors. It may for example vary with the orientation of the patch. In many cases, however, it

remains at least approximately constant as the surface moves in the environment. If we assume that the image brightness of a patch remains constant, we have

$$\frac{dE}{dt} = 0,$$

or

$$\frac{\partial E}{\partial \mathbf{r}} \cdot \frac{d\mathbf{r}}{dt} + \frac{\partial E}{\partial t} = 0,$$

where $\partial E/\partial \mathbf{r} = (\partial E/\partial x, \partial E/\partial y, 0)^T$ is the image brightness gradient. It is convenient to use the notation $E_\mathbf{r}$ for this quantity and $E_t$ for the time derivative of the brightness. Then we can write the brightness change equation in the simple form

$$E_\mathbf{r} \cdot \mathbf{r}_t + E_t = 0.$$

Substituting for $\mathbf{r}_t$ we get

$$E_t - E_\mathbf{r} \cdot \Big(\hat{\mathbf{z}} \times \big(\mathbf{r} \times (\mathbf{r} \times \omega - \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}}\mathbf{t})\big)\Big) = 0.$$

Now

$$E_\mathbf{r} \cdot \big(\hat{\mathbf{z}} \times (\mathbf{r} \times \mathbf{t})\big) = (E_\mathbf{r} \times \hat{\mathbf{z}}) \cdot (\mathbf{r} \times \mathbf{t}) = \big((E_\mathbf{r} \times \hat{\mathbf{z}}) \times \mathbf{r}\big) \cdot \mathbf{t},$$

and by similar reasoning

$$E_\mathbf{r} \cdot \Big(\hat{\mathbf{z}} \times \big(\mathbf{r} \times (\mathbf{r} \times \omega)\big)\Big) = \Big(\big((E_\mathbf{r} \times \hat{\mathbf{z}}) \times \mathbf{r}\big) \times \mathbf{r}\Big) \cdot \omega,$$

so we have

$$E_t - \Big(\big((E_\mathbf{r} \times \hat{\mathbf{z}}) \times \mathbf{r}\big) \times \mathbf{r}\Big) \cdot \omega + \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}}\big((E_\mathbf{r} \times \hat{\mathbf{z}}) \times \mathbf{r}\big) \cdot \mathbf{t} = 0.$$

To make this constraint equation more compact, let us define $c = E_t$, $\mathbf{s} = (E_\mathbf{r} \times \hat{\mathbf{z}}) \times \mathbf{r}$, and $\mathbf{v} = -\mathbf{s} \times \mathbf{r}$; then, finally,

$$c + \mathbf{v} \cdot \omega + \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}}\mathbf{s} \cdot \mathbf{t} = 0.$$

This is the brightness change equation in the case of rigid body motion.

## 2.5. Planar Surface

A particularly impoverished scene is one consisting of a single planar surface. The equation for such a surface is

$$\mathbf{R} \cdot \mathbf{n} = 1,$$

where $\mathbf{n}/|\mathbf{n}|$ is a unit normal to the plane, and $1/|\mathbf{n}|$ is the perpendicular distance of the plane from the origin. Since $\mathbf{R} = (\mathbf{R} \cdot \hat{\mathbf{z}})\mathbf{r}$, we can write this as

$$\mathbf{r} \cdot \mathbf{n} = \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}},$$

so the constraint equation becomes

$$c + \mathbf{v} \cdot \omega + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0.$$

This is the brightness change equation for a planar surface.

Note again the inherent ambiguity in the constraint equation. It is satisfied equally well by two planes with the same orientation but at different distances provided that the translational velocities are in the same proportions.

## 3. Recovering Motion and Structure

Given image brightness $E(x, y, t)$, and its spatial and time derivatives, $E_{\mathbf{r}}$ and $E_t$, over some region $I$ in the image plane, we are to recover the translational and rotational motions, $\mathbf{t}$ and $\omega$, as well as the plane $\mathbf{n}$. Using the constraint equation developed above, we could do this using image information at just a small number of points. At each point we get one constraint and we have nine unknowns to recover—or rather, eight, since we can recover the distance of the plane and the translational velocity only up to a scale factor.

### 3.1. Least Squares Approach

Image brightness values are distorted with sensor noise and quantization error. These inaccuracies are further accentuated by methods used for estimating the brightness gradient. Thus it is not advisable to base a method on measurements at just a few points. Instead we propose to minimize the error in the brightness constraint equation over the whole region $I$ in the image plane. So we wish to minimize

$$J = \iint_I [c + \mathbf{v} \cdot \omega + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t})]^2 \, dx \, dy$$

by suitable choice of the translational and rotational motion vectors $\mathbf{t}$ and $\omega$, as well as the normal to the plane, $\mathbf{n}$. We are particularly interested in the question of uniqueness of the solution and in computational efficiency.

For an extremum of $J$ we must have

$$\frac{\partial J}{\partial \omega} = \mathbf{0}, \quad \frac{\partial J}{\partial \mathbf{t}} = \mathbf{0}, \qquad \text{and} \qquad \frac{\partial J}{\partial \mathbf{n}} = \mathbf{0}.$$

That is,

$$\iint_I [c + \mathbf{v} \cdot \omega + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t})] \, \mathbf{v} \, dx \, dy = \mathbf{0},$$

$$\iint_I (\mathbf{r} \cdot \mathbf{n}) \, [c + \mathbf{v} \cdot \omega + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t})] \, \mathbf{s} \, dx \, dy = \mathbf{0},$$

$$\iint_I (\mathbf{s} \cdot \mathbf{t}) \, [c + \mathbf{v} \cdot \omega + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t})] \, \mathbf{r} \, dx \, dy = \mathbf{0}.$$

These equations comprise nine non-linear (scalar) algebraic equation in terms of the observer motion, $\mathbf{t}$ and $\omega$, and the surface normal $\mathbf{n}$.

## 4. Iterative Solution Methods

To recover the motion vectors $\mathbf{t}$ and $\omega$, and the surface normal $\mathbf{n}$, we have to solve a set of non-linear equations, which we will call the *planar motion field equations.* Some observations about these equations are in order: The first equation is linear in $\omega$, $\mathbf{t}$, and $\mathbf{n}$. The second equation is linear in $\omega$ and $\mathbf{t}$, but quadratic in $\mathbf{n}$. Finally, the last equation is linear in $\omega$ and $\mathbf{n}$, but quadratic in $\mathbf{t}$. We will exploit the linearity of these equations to formulate two iterative schemes.

### 4.1. First Scheme

We can rearrange the motion and surface recovery equations to get

$$\left[\iint_I (\mathbf{v}\mathbf{v}^T)\,dx\,dy\right]\omega + \left[\iint_I (\mathbf{r}\cdot\mathbf{n})(\mathbf{v}\mathbf{s}^T)\,dx\,dy\right]\mathbf{t} = -\iint_I c\mathbf{v}\,dx\,dy,$$

$$\left[\iint_I (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\mathbf{v}^T)\,dx\,dy\right]\omega + \left[\iint_I (\mathbf{r}\cdot\mathbf{n})^2(\mathbf{s}\mathbf{s}^T)\,dx\,dy\right]\mathbf{t} = -\iint_I c(\mathbf{r}\cdot\mathbf{n})\mathbf{s}\,dx\,dy,$$

$$\left[\iint_I (\mathbf{s}\cdot\mathbf{t})^2(\mathbf{r}\mathbf{r}^T)\,dx\,dy\right]\mathbf{n} = -\iint_I [c + (\mathbf{v}\cdot\mathbf{w})](\mathbf{s}\cdot\mathbf{t})\mathbf{r}\,dx\,dy,$$

The first pair can be grouped in the form

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix}\begin{pmatrix} \omega \\ \mathbf{t} \end{pmatrix} = -\begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix},$$

where

$$\mathbf{M}_1 = \iint_I (\mathbf{v}\mathbf{v}^T)\,dx\,dy, \quad \mathbf{M}_2 = \iint_I (\mathbf{r}\cdot\mathbf{n})(\mathbf{v}\mathbf{s}^T)\,dx\,dy, \quad \mathbf{M}_4 = \iint_I (\mathbf{r}\cdot\mathbf{n})^2(\mathbf{s}\mathbf{s}^T)\,dx\,dy,$$

$$\mathbf{d}_1 = \iint_I c\mathbf{v}\,dx\,dy, \qquad \text{and} \qquad \mathbf{d}_2 = \iint_I c(\mathbf{r}\cdot\mathbf{n})\mathbf{s}\,dx\,dy.$$

This can be solved for $\mathbf{t}$ and $\omega$, given the surface normal $\mathbf{n}$. The last equation is

$$\mathbf{N}_4\mathbf{n} = -\mathbf{g},$$

where

$$\mathbf{N}_4 = \iint_I (\mathbf{s}\cdot\mathbf{t})^2(\mathbf{r}\mathbf{r}^T)\,dx\,dy,$$

$$\mathbf{g} = \iint_I [c + (\mathbf{v}\cdot\omega)]\,(\mathbf{s}\cdot\mathbf{t})\mathbf{r}\,dx\,dy,$$

and can be solved for the surface normal $\mathbf{n}$, given the pair of vectors $\mathbf{t}$ and $\omega$.

The motion vectors are given by

$$\omega = \left(\mathbf{M}_2^{-1}\mathbf{M}_1 - \mathbf{M}_4^{-1}\mathbf{M}_2^T\right)^{-1}\left(\mathbf{M}_4^{-1}\mathbf{d}_2 - \mathbf{M}_2^{-1}\mathbf{d}_1\right),$$

$$\mathbf{t} = \left(\mathbf{M}_1^{-1}\mathbf{M}_2 - \mathbf{M}_2^{-T}\mathbf{M}_4\right)^{-1}\left(\mathbf{M}_2^{-T}\mathbf{d}_2 - \mathbf{M}_1^{-1}\mathbf{d}_1\right),$$

where $(^{-T})$ denotes the inverse of the transpose of a matrix. This can also be written in the form

$$\mathbf{t} = \left(\mathbf{M}_4 - \mathbf{M}_2^T\mathbf{M}_1^{-1}\mathbf{M}_2\right)^{-1}\left(\mathbf{M}_2^T\mathbf{M}_1^{-1}\mathbf{d}_1 - \mathbf{d}_2\right),$$

$$\boldsymbol{\omega} = -\mathbf{M}_1^{-1}\left(\mathbf{d}_1 + \mathbf{M}_2\mathbf{t}\right).$$

or,

$$\boldsymbol{\omega} = \left(\mathbf{M}_1 - \mathbf{M}_2\mathbf{M}_4^{-1}\mathbf{M}_2^T\right)^{-1}\left(\mathbf{M}_2\mathbf{M}_4^{-1}\mathbf{d}_2 - \mathbf{d}_1\right),$$

$$\mathbf{t} = -\mathbf{M}_4^{-1}\left(\mathbf{d}_2 + \mathbf{M}_2^T\boldsymbol{\omega}\right),$$

The surface normal is simply given by

$$\mathbf{n} = -\mathbf{N}_4^{-1}\mathbf{g}.$$

All arrays are either $3 \times 3$ matrices or vectors of length 3, and therefore, the solutions for $\boldsymbol{\omega}$, $\mathbf{t}$, and $\mathbf{n}$ can be computed easily. Actually, most of the indicated matrix inversions do not have to be carried out explicitly, since it is computationally cheaper to solve these linear matrix equations by elimination.

So in summary: We start with an initial guess for $\mathbf{n}$. Using above equations, we solve for $\mathbf{t}$ and $\boldsymbol{\omega}$ in terms of the current value of $\mathbf{n}$, and then for $\mathbf{n}$ in terms of the current values of $\mathbf{t}$ and $\boldsymbol{\omega}$. After this, we evaluate the improvement in the solution to either go to next iteration or stop if the solution has not improved.

### 4.2. Second Scheme

The first pair of the motion and surface recovery equations depend linearly on $\mathbf{t}$ and $\boldsymbol{\omega}$. As before,

$$\left[\iint_I (\mathbf{v}\mathbf{v}^T)\,dx\,dy\right]\boldsymbol{\omega} + \left[\iint_I (\mathbf{r}\cdot\mathbf{n})(\mathbf{v}\mathbf{s}^T)\,dx\,dy\right]\mathbf{t} = -\iint_I c\mathbf{v}\,dx\,dy,$$

$$\left[\iint_I (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\mathbf{v}^T)\,dx\,dy\right]\boldsymbol{\omega} + \left[\iint_I (\mathbf{r}\cdot\mathbf{n})^2(\mathbf{s}\mathbf{s}^T)\,dx\,dy\right]\mathbf{t} = -\iint_I c(\mathbf{r}\cdot\mathbf{n})\mathbf{s}\,dx\,dy,$$

which can be solved for $\mathbf{t}$ and $\boldsymbol{\omega}$ in terms of $\mathbf{n}$. Furthermore, the first and last equations depend linearly on $\mathbf{n}$ and $\boldsymbol{\omega}$:

$$\left[\iint_I (\mathbf{v}\mathbf{v}^T)\,dx\,dy\right]\boldsymbol{\omega} + \left[\iint_I (\mathbf{s}\cdot\mathbf{t})(\mathbf{v}\mathbf{r}^T)\,dx\,dy\right]\mathbf{n} = -\iint_I c\mathbf{v}\,dx\,dy,$$

$$\left[\iint_I (\mathbf{s}\cdot\mathbf{t})(\mathbf{r}\mathbf{v}^T)\,dx\,dy\right]\boldsymbol{\omega} + \left[\iint_I (\mathbf{s}\cdot\mathbf{t})^2(\mathbf{r}\mathbf{r}^T)\,dx\,dy\right]\mathbf{n} = -\iint_I c(\mathbf{s}\cdot\mathbf{t})\mathbf{r}\,dx\,dy.$$

Given $\mathbf{t}$, these may be solved for $\mathbf{n}$ and $\boldsymbol{\omega}$. For simplicity, let $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_4$, $\mathbf{N}_4$, $\mathbf{d}_1$, and $\mathbf{d}_2$ be as defined earlier, and let:

$$\mathbf{N}_1 = \mathbf{M}_1, \qquad \mathbf{d}_1 = \mathbf{e}_1,$$

$$\mathbf{N}_2 = \iint_I (\mathbf{s}\cdot\mathbf{t})(\mathbf{v}\mathbf{r}^T)\,dx\,dy, \quad \text{and} \quad \mathbf{e}_2 = \iint_I c(\mathbf{s}\cdot\mathbf{t})\mathbf{r}\,dx\,dy.$$

Then

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix} \begin{pmatrix} \omega \\ \mathbf{t} \end{pmatrix} = - \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix},$$

and

$$\begin{pmatrix} \mathbf{N}_1 & \mathbf{N}_2 \\ \mathbf{N}_2^T & \mathbf{N}_4 \end{pmatrix} \begin{pmatrix} \omega \\ \mathbf{n} \end{pmatrix} = - \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix}.$$

The solution of the above equations is given by

$$\omega = \left(\mathbf{M}_2^{-1}\mathbf{M}_1 - \mathbf{M}_4^{-1}\mathbf{M}_2^T\right)^{-1} \left(\mathbf{M}_4^{-1}\mathbf{d}_2 - \mathbf{M}_2^{-1}\mathbf{d}_1\right),$$

$$\mathbf{t} = \left(\mathbf{M}_1^{-1}\mathbf{M}_2 - \mathbf{M}_2^{-T}\mathbf{M}_4\right)^{-1} \left(\mathbf{M}_2^{-T}\mathbf{d}_2 - \mathbf{M}_1^{-1}\mathbf{d}_1\right),$$

and

$$\omega = \left(\mathbf{N}_2^{-1}\mathbf{N}_1 - \mathbf{N}_4^{-1}\mathbf{N}_2^T\right)^{-1} \left(\mathbf{N}_4^{-1}\mathbf{e}_2 - \mathbf{N}_2^{-1}\mathbf{e}_1\right),$$

$$\mathbf{n} = \left(\mathbf{N}_1^{-1}\mathbf{N}_2 - \mathbf{N}_2^{-T}\mathbf{N}_4\right)^{-1} \left(\mathbf{N}_2^{-T}\mathbf{e}_2 - \mathbf{N}_1^{-1}\mathbf{e}_1\right),$$

These may be rewritten in either of two asymmetrical forms as shown above.

Again, most of the indicated matrix inversions do not have to be carried out explicitly, since we can solve the equations by elimination.

In this scheme: We start with an initial guess for $\mathbf{n}$. We solve for $\mathbf{t}$ and $\omega$ in terms of the current value of $\mathbf{n}$, and update $\mathbf{t}$, then solve for $\mathbf{n}$ and $\omega$ in terms of the current value of $\mathbf{t}$, and update $\mathbf{n}$, and, finally, evaluate the improvement in the solution to either continue with the next iteration or stop if the solution has not improved.

### 4.3. Division of Labour

These methods would not be very attractive, if we had to perform integrations over the whole image region $I$ during each iteration, in order to collect the matrices and vectors appearing in the equations. Fortunately, this is not neccessary. One can see this by writing the equations for the components of the matrices and vectors using the summation convention of tensor calculus (that is, there is an implicit summation over any index which appears twice in an expression):

$$\{\mathbf{M}_1\}_{i,j} = \iint_I v_i v_j \, dx\, dy,$$

$$\{\mathbf{M}_2\}_{i,j} = \left[\iint_I v_i s_j r_k \, dx\, dy\right] n_k,$$

$$\{\mathbf{M}_4\}_{i,j} = \left[\iint_I s_i s_j r_k r_l \, dx\, dy\right] n_k n_l,$$

$$\{\mathbf{d}_1\}_i = \iint_I c v_i \, dx\, dy, \qquad \text{and} \qquad \{\mathbf{d}_2\}_i = \left[\iint_I c s_i r_j \, dx\, dy\right] n_j,$$

$$\{\mathbf{N}_1\}_{i,j} = \iint_I v_i v_j \, dx \, dy,$$

$$\{\mathbf{N}_2\}_{i,j} = \left[\iint_I v_i s_k r_j \, dx \, dy\right] t_k,$$

$$\{\mathbf{N}_4\}_{i,j} = \left[\iint_I s_k s_l r_i r_j \, dx \, dy\right] t_k t_l,$$

$$\{\mathbf{e}_1\}_i = \iint_I c v_i \, dx \, dy, \qquad \text{and} \qquad \{\mathbf{e}_2\}_i = \left[\iint_I c s_j r_i \, dx \, dy\right] t_j,$$

and

$$\{\mathbf{g}\}_i = \left[\iint_I c r_i s_j \, dx \, dy\right] t_j + \left[\iint_I s_k r_i v_j \, dx \, dy\right] t_k w_j.$$

$\mathbf{M}_1 = \mathbf{N}_1$ and $\mathbf{d}_1 = \mathbf{e}_1$ do not depend on $\omega$, $\mathbf{t}$, or $\mathbf{n}$, and so need only be computed once. Also, $(cv_i)$, $(v_i v_j)$, $(cs_i r_j)$, $(r_k v_i s_j)$, and $(r_k r_l s_i s_j)$ depend only on $\mathbf{r}$, $E_\mathbf{r}$, and $E_t$, and so can be integrated over the image once. This appears to be a set of $3+9+9+27+81 = 129$ numbers, but, because of symmetry in $(v_i v_j)$, and $(r_k r_l s_i s_j)$, only 81 numbers have to be stored. These accumulated totals represent all the image information needed to solve the motion recovery problem.

In the first scheme, we only perform 279 multiplications per iteration; The updating of the coefficients of the planar motion field equations involves $27+9+42+42+42 = 162$ multiplications to compute $\mathbf{M}_2$, $\mathbf{d}_2$, $\mathbf{M}_4$, $\mathbf{N}_4$, and $\mathbf{g}$ (note that $\mathbf{M}_4$ and $\mathbf{N}_4$ are symmetric). The updating of $\omega$, $\mathbf{t}$, and $\mathbf{n}$, in comparison, requires 117 multiplications.

In the second scheme, 696 multiplications are carried out at each iteration; We compute the matrices $\mathbf{M}_2$, $\mathbf{M}_4$ and the vector $\mathbf{d}_2$, required for the first half of the iteration, in $27 + 42 + 9 = 78$ multiplications. The same number of multiplications is needed to compute the matrices $\mathbf{N}_2$, $\mathbf{N}_4$ and the vector $\mathbf{e}_2$ required in the second half. Further, solving for $\omega$ and $\mathbf{t}$ takes about 270 multiplications, as does solving for $\omega$ and $\mathbf{n}$ in the second half of each iterative step.

Through a selected example, we will show that the second scheme has a much better convergence rate at the expense of more computation per iteration.

## 5. Uniqueness

It is important to establish whether more than one solution is possible. In general, this is clearly so, since an image of uniform brightness could correspond to an arbitrary, uniform surface moving in an arbitrary way. So the brightness gradients, or lack of brightness gradients, can conspire to make the problem highly ambiguous. What we are interested in here is whether two different planar surfaces can give rise to the same motion field given two different translational and rotational motions of the imaging system.

In our terms then, the question becomes: given that the brightness change equation is satisfied for the motion $\mathbf{t}$ and $\omega$ and the planar surface $\mathbf{n}$, is there another motion $\mathbf{t}'$ and $\omega'$ and another planar surface $\mathbf{n}'$ that satisfies the same equation at all points in the region $I$ and for all possibly ways of marking the surface? Note that we have to consider a whole image region, since the problem is underconstrained if we only have

information along a line or at a point in the image. We also have to include the condition that the constraint should be satisfied for all possible surface markings to avoid the kind of ambiguity discussed above, where brightness gradients fortuitously line up with the motion field to create ambiguity.

### 5.1. Dual Solution

Suppose that two motions and two planar surfaces satisfy the brightness change equation. Then, we have

$$c + \mathbf{v} \cdot \omega + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0,$$
$$c + \mathbf{v} \cdot \omega' + (\mathbf{r} \cdot \mathbf{n}')(\mathbf{s} \cdot \mathbf{t}') = 0.$$

Subtracting these equations, we get

$$\mathbf{v} \cdot (\omega - \omega') + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) - (\mathbf{r} \cdot \mathbf{n}')(\mathbf{s} \cdot \mathbf{t}') = 0.$$

Now $\mathbf{v} = -\mathbf{s} \times \mathbf{r}$, so

$$-(\mathbf{s} \times \mathbf{r}) \cdot (\omega - \omega') + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) - (\mathbf{r} \cdot \mathbf{n}')(\mathbf{s} \cdot \mathbf{t}') = 0,$$

or

$$-\mathbf{r} \cdot \big((\omega - \omega') \times \mathbf{s}\big) + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) - (\mathbf{r} \cdot \mathbf{n}')(\mathbf{s} \cdot \mathbf{t}') = 0.$$

If we let $\omega = (\omega_1, \omega_2, \omega_3)^T$, then we can write

$$\omega \times \mathbf{s} = \Omega \mathbf{s}, \quad \text{where} \quad \Omega = \begin{pmatrix} 0 & -\omega_3 & +\omega_2 \\ +\omega_3 & 0 & -\omega_1 \\ -\omega_2 & +\omega_1 & 0 \end{pmatrix},$$

is a suitable $(3 \times 3)$ skew-symmetric matrix. The $(i,j)^{\text{th}}$ element of $\Omega$ equals $\omega_k \epsilon_{ijk}$, where $\epsilon_{ijk}$, is the permutation symbol. (It equals $+1$ when the ordered set $i$, $j$, $k$ is obtained by an even permutation of the set 1, 2, and 3, it equals $-1$ when the ordered set is obtained by an odd permutation, and it is zero if two or more of the indices are equal.)

Using this notation we can now write

$$-\mathbf{r}^T(\Omega - \Omega')\mathbf{s} + \mathbf{r}^T(\mathbf{n}\mathbf{t}^T)\mathbf{s} - \mathbf{r}^T(\mathbf{n}'\mathbf{t}'^T)\mathbf{s} = 0,$$

or just

$$\mathbf{r}^T \left[-(\Omega - \Omega') + \mathbf{n}\mathbf{t}^T - \mathbf{n}'\mathbf{t}'^T\right] \mathbf{s} = 0.$$

This is to be true for all points $\mathbf{r}$ in the image region $I$ and all possible brightness gradients. So

$$-(\Omega - \Omega') + \mathbf{n}\mathbf{t}^T - \mathbf{n}'\mathbf{t}'^T = \mathbf{0},$$

where the zero on the right hand side here represents a $3 \times 3$ matrix of zeros. Now $\Omega^T = -\Omega$, since $\Omega$ is skew-symmetric, so taking the transpose of the equation we get

$$+(\Omega - \Omega') + \mathbf{t}\mathbf{n}^T - \mathbf{t}'\mathbf{n}'^T = 0.$$

Adding the two equations allows us to eliminate $(\Omega - \Omega')$, and we end up with

$$(\mathbf{n}\mathbf{t}^T + \mathbf{t}\mathbf{n}^T) = (\mathbf{n}'\mathbf{t}'^T + \mathbf{t}'\mathbf{n}'^T).$$

The trace (sum of the diagonal elements) of $\mathbf{n}\mathbf{t}^T$ is just $(\mathbf{n} \cdot \mathbf{t})$, so we see immediately that $(\mathbf{n} \cdot \mathbf{t}) = (\mathbf{n}' \cdot \mathbf{t}'^T)$. But the above matrix equation involving the dyadic products of $\mathbf{n}$ and $\mathbf{t}$ as well as of $\mathbf{n}'$ and $\mathbf{t}'$ is much more constraining.

Consider the following three possibilities:

(1) If $|\mathbf{n}'| = 0$ or $|\mathbf{t}'| = 0$, then their dyadic product is a $3 \times 3$ matrix of zeros. In this case the above equation is satisfied if and only if $|\mathbf{n}| = 0$ or $|\mathbf{t}| = 0$.

(2) If $\mathbf{n}' \parallel \mathbf{n}$, $\mathbf{t}' \parallel \mathbf{t}$ and $|\mathbf{n}'|\,|\mathbf{t}'| = |\mathbf{n}|\,|\mathbf{t}|$, then the two sums of dyadic products are equal and the above equation is satisfied.

(3) If $\mathbf{n}' \parallel \mathbf{t}$, $\mathbf{t}' \parallel \mathbf{n}$ and $|\mathbf{n}'|\,|\mathbf{t}'| = |\mathbf{n}|\,|\mathbf{t}|$, then the two sums of dyadic products are also equal and the above equation is satisfied.

It turns out that there are no other ways to satisfy the equation. This can be shown using elementary properties of dyadic products (see Appendix) or by inspection of the six components of the above equation (because of symmetry there are only six independent components).

The first case above corresponds to purely rotational motion, because either the translational motion is zero, or the planar surface is infinitely far away, and the translation does not generate a perceptible component of the motion field. The solution is unique in this case, because we find $(\Omega - \Omega') = \mathbf{0}$, when we substitute back into the matrix equation. (This is nothing new, since it has been known for some time that the solution is unique in the case of purely rotational and purely translational motion [3].)

In the second case we find that $\mathbf{n}\mathbf{t}^T = \mathbf{n}'\mathbf{t}'^T$, since the vectors are parallel and the product of their size is constrained by the condition $\mathbf{n} \cdot \mathbf{t} = \mathbf{n}' \cdot \mathbf{t}'$, derived earlier. Thus once again $(\Omega - \Omega') = \mathbf{0}$. Nothing new is obtained here, since we already know that we can change the lengths of the vectors $\mathbf{n}$ and $\mathbf{t}$ as long as the product of their lengths remains constant.

The third case is the most interesting. Here we have $\mathbf{t}\mathbf{n}^T = \mathbf{n}'\mathbf{t}'^T$ so that

$$-(\Omega - \Omega') + (\mathbf{n}\mathbf{t}^T - \mathbf{t}\mathbf{n}^T) = \mathbf{0},$$

and thus

$$-(\Omega - \Omega')\mathbf{x} + (\mathbf{n}\mathbf{t}^T - \mathbf{t}\mathbf{n}^T)\mathbf{x} = \mathbf{0},$$

for an arbitrary vector $\mathbf{x}$. That is,

$$\mathbf{x} \times (\omega - \omega') + \mathbf{x} \times (\mathbf{n} \times \mathbf{t}) = \mathbf{0},$$

for an arbitrary vector $\mathbf{x}$, so that

$$\omega - \omega' + \mathbf{n} \times \mathbf{t} = \mathbf{0},$$

or $\omega' = \omega + \mathbf{n} \times \mathbf{t}$. To summarize then, if we ignore scaling of the normal and the translational velocity, we obtain a dual solution, given by

$$\mathbf{n}' = \mathbf{t}, \quad \mathbf{t}' = \mathbf{n} \quad \text{and} \quad \omega' = \omega + \mathbf{n} \times \mathbf{t}.$$

Hay was the first to show the existence of the dual solution [6], although the result has apparently been independently rediscovered several times since then [11,13,15]. (The most recent of these papers [11], came to our attention only after completion of our version of the proof.)

This dual solution is not different from the original one in the special case that the motion is perpendicular to the planar surface, that is, $\mathbf{n} \parallel \mathbf{t}$. In this case the solution is unique. Further, if $\mathbf{t} \cdot \hat{\mathbf{z}} = 0$, then $\mathbf{n}' \cdot \hat{\mathbf{z}} = 0$. This corresponds to a planar surface parallel to the observer's line of sight, and may be considered to be a degenerate case.

## 6. A Selected Example

We now present the results of a simulation. It is noteworthy to mention that in all simulations performed, our algorithms have converged to a solution. However, the number of iterations for convergence to a solution depends on the initial condition (as is the case with all iterative schemes developed for solving nonlinear equations). In this example, we will demonstrate the sensitivity of both schemes to the initial condition. The image brightness function was generated using a multiplicative sinusoidal pattern (one that varies sinusoidally in both $x$ and $y$ directions), a 45° field of view was assumed, and the image brightness gradients were computed analytically to avoid errors due to image brightness quantization and finite difference approximations of the brightness gradient. In practice, the brightness at image points in two frames would be discretized first, and the gradient computed using finite difference methods.

Table 1 shows the true motion and surface parameters, and the results of a simulation that converged to the true solution using the first scheme described earlier. In Table 2, the dual solution for the true motion and surface parameters, and the results of a simulation that converged to the dual solution are tabulated. In both cases, the solutions after various number of iterations are given. The results show that in the first case, the error in each parameter after less than 30 iterations is within 10% of the exact value. In the second case, this accuracy is achieved in less than 20 iterations. Similar results are presented in tables 3 and 4 for the second scheme. Here, very good accuracy is achieved in less that 10 iterations for the true solution and about 5 iterations for the dual solution.

In similar tests, with various motion and surface parameters, accurate results have been obtained in less than 40 iterations using the first scheme and a variety of initial conditions. The same accuracy for the second scheme required less than 15 iterations. More importantly, both schemes eventually converged to one of the two possible solutions. However, the results for the particular case where the translational motion vector is (almost) parallel to the surface normal have not been as satisfactory. In these cases, several hundred iterations were required to achieve reasonable accuracy, even with the second scheme. Although the nature of this behavior has not been investigated in detail, it appears to resemble that observed when the Newton-Raphson method is applied to a problem where two roots are very close to one another.

Table 1: The True Motion and Surface Parameters, and a Summary of the Results of a Simulation That Converges to the True Solution.

| | | | |
|---|---|---|---|
| True Rotational Motion Parameters | $w_1 = .003$ | $w_2 = .001$ | $w_3 = -.01$ |
| True Translational Motion Parameters | $t_1 = .0005$ | $t_2 = -.005$ | $t_3 = .0125$ |
| True Parameters of the Surface | $n_1 = .2$ | $n_2 = .4$ | $n_3 = 1.0$ |
| Initial Guess for the Simulation | $n_1 = 100.$ | $n_2 = 5.$ | $n_3 = -1.$ |

| Iter. | (Rotational Par's) | | | (Translational Par's) | | | (Surface Par's) | | |
|---|---|---|---|---|---|---|---|---|---|
| No. | $w_1$ | $w_2$ | $w_3$ | $t_1$ | $t_2$ | $t_3$ | $n_1$ | $n_2$ | $n_3$ |
| 10 | .00531 | .00260 | -.01016 | -.00069 | -.00284 | .01301 | .35524 | .1923 | 1. |
| 15 | .00429 | .00178 | -.01008 | -.00006 | -.00384 | .01291 | .27623 | .2742 | 1. |
| 20 | .00353 | .00137 | -.01002 | .00024 | -.00454 | .01270 | .23725 | .3448 | 1. |
| 25 | .00318 | .00117 | -.01000 | .00038 | -.00485 | .01257 | .21718 | .3814 | 1. |
| 30 | .00305 | .00107 | -.01000 | .00045 | -.00495 | .01252 | .20755 | .3945 | 1. |
| 35 | .00302 | .00103 | -.01000 | .00048 | -.00499 | .01250 | .20323 | .3984 | 1. |
| 40 | .00300 | .00101 | -.01000 | .00049 | -.00500 | .01250 | .20137 | .3996 | 1. |
| 45 | .00300 | .00101 | -.01000 | .00050 | -.00500 | .01250 | .20058 | .3999 | 1. |
| 50 | .00300 | .00100 | -.01000 | .00050 | -.00500 | .01250 | .20024 | .4000 | 1. |
| 55 | .00300 | .00100 | -.01000 | .00050 | -.00500 | .01250 | .20010 | .4000 | 1. |
| 60 | .00300 | .00100 | -.01000 | .00050 | -.00500 | .01250 | .20004 | .4000 | 1. |
| 65 | .00300 | .00100 | -.01000 | .00050 | -.00500 | .01250 | .20002 | .4000 | 1. |
| 70 | .00300 | .00100 | -.01000 | .00050 | -.00500 | .01250 | .20000 | .4000 | 1. |

Table 2: The Dual Motion and Surface Parameters, and a Summary of the Results of a Simulation That Converges to the Dual Solution Using the First Scheme.

| | | | |
|---|---|---|---|
| Dual Rotational Motion Parameters | $w_1 = .013$ | $w_2 = -.001$ | $w_3 = -.0112$ |
| Dual Translational Motion Parameters | $t_1 = .0025$ | $t_2 = .005$ | $t_3 = .0125$ |
| Parameters of the Dual Surface | $n_1 = .04$ | $n_2 = -.4$ | $n_3 = 1.0$ |
| Initial Guess for the Simulation | $n_1 = .5$ | $n_2 = 1.5$ | $n_3 = -1.$ |

| Iter. | (Rotational Par's) | | | (Translational Par's) | | | (Surface Par's) | | |
|---|---|---|---|---|---|---|---|---|---|
| No. | $w_1$ | $w_2$ | $w_3$ | $t_1$ | $t_2$ | $t_3$ | $n_1$ | $n_2$ | $n_3$ |
| 10 | .01302 | -.00120 | -.01118 | .00266 | .00503 | .01247 | .01941 | -.4021 | 1. |
| 15 | .01299 | -.00108 | -.01119 | .00256 | .00500 | .01249 | .03220 | -.3992 | 1. |
| 20 | .01299 | -.00103 | -.01120 | .00253 | .00500 | .01250 | .03692 | -.3993 | 1. |
| 25 | .01300 | -.00101 | -.01120 | .00251 | .00500 | .01250 | .03876 | -.3996 | 1. |
| 30 | .01300 | -.00101 | -.01120 | .00250 | .00500 | .01250 | .03950 | -.3998 | 1. |
| 35 | .01300 | -.00100 | -.01120 | .00250 | .00500 | .01250 | .03980 | -.3999 | 1. |
| 40 | .01300 | -.00100 | -.01120 | .00250 | .00500 | .01250 | .03992 | -.4000 | 1. |
| 45 | .01300 | -.00100 | -.01120 | .00250 | .00500 | .01250 | .03997 | -.4000 | 1. |
| 50 | .01300 | -.00100 | -.01120 | .00250 | .00500 | .01250 | .03999 | -.4000 | 1. |

Table 3: The True Motion and Surface Parameters, and a Summary of the Results of a Simulation That Converges to the True Solution Using the Second Scheme.

| | | | |
|---|---|---|---|
| True Rotational Motion Parameters | $w_1 = .003$ | $w_2 = .001$ | $w_3 = -.01$ |
| True Translational Motion Parameters | $t_1 = .0005$ | $t_2 = -.005$ | $t_3 = .0125$ |
| True Parameters of the Surface | $n_1 = .2$ | $n_2 = .4$ | $n_3 = 1.0$ |
| Initial Guess for the Simulation | $n_1 = 100.$ | $n_2 = 5.$ | $n_3 = -1.$ |

| Iter. | (Rotational Par's) | | | (Translational Par's) | | | (Surface Par's) | | |
|---|---|---|---|---|---|---|---|---|---|
| No. | $w_1$ | $w_2$ | $w_3$ | $t_1$ | $t_2$ | $t_3$ | $n_1$ | $n_2$ | $n_3$ |
| 5 | .00254 | .00105 | -.00990 | .00039 | -.00546 | .01202 | .20581 | .44259 | 1. |
| 10 | .00296 | .00100 | -.00999 | .00049 | -.00504 | .01246 | .20039 | .40375 | 1. |
| 15 | .00300 | .00100 | -.01000 | .00050 | -.00500 | .01250 | .20004 | .40037 | 1. |
| 20 | .00300 | .00100 | -.01000 | .00050 | -.00500 | .01250 | .20000 | .40004 | 1. |
| 25 | .00300 | .00100 | -.01000 | .00050 | -.00500 | .01250 | .20000 | .40000 | 1. |

Table 4: The Dual Motion and Surface Parameters, and a Summary of the Results of a Simulation That Converges to the Dual Solution Using the Second Scheme.

| | | | |
|---|---|---|---|
| Dual Rotational Motion Parameters | $w_1 = .013$ | $w_2 = -.001$ | $w_3 = -.0112$ |
| Dual Translational Motion Parameters | $t_1 = .0025$ | $t_2 = .005$ | $t_3 = .0125$ |
| Parameters of the Dual Surface | $n_1 = .04$ | $n_2 = -.4$ | $n_3 = 1.0$ |
| Initial Guess for the Simulation | $n_1 = .5$ | $n_2 = 1.5$ | $n_3 = -1.$ |

| Iter. | (Rotational Par's) | | | (Translational Par's) | | | (Surface Par's) | | |
|---|---|---|---|---|---|---|---|---|---|
| No. | $w_1$ | $w_2$ | $w_3$ | $t_1$ | $t_2$ | $t_3$ | $n_1$ | $n_2$ | $n_3$ |
| 5 | .01330 | -.00102 | -.01122 | .00248 | .00531 | .01221 | .03790 | -.42663 | 1. |
| 10 | .01303 | -.00100 | -.01120 | .00250 | .00503 | .01248 | .03979 | -.40245 | 1. |
| 15 | .01300 | -.00100 | -.01120 | .00250 | .00500 | .01250 | .03998 | -.40024 | 1. |
| 20 | .01300 | -.00100 | -.01120 | .00250 | .00500 | .01250 | .04000 | -.40002 | 1. |
| 25 | .01300 | -.00100 | -.01120 | .00250 | .00500 | .01250 | .04000 | -.40000 | 1. |

## 7. Summary

The problem of recovering the motion of an observer relative to a planar surface directly from the changing images (direct passive navigation) was investigated, and formulated as one of unconstrained optimization. Using conditions for optimality, it was reduced to solving a set of nine simultaneous non-linear equations, which we termed the *planar motion field equations*. Two iterative schemes for solving these equations were presented.

It was shown that all information in the image concerning motion recovery can be captured by the moments of the image brightness derivatives which constitute the coefficients of the motion and surface recovery equations. These moments are computed during an initial pass over the relevant image regions so that there is no need to refer to the image after every iteration. This reduces the computation to storing 81 moments and performing less than 300 multiplications per iteration in the first scheme and approximately 700 multiplications in the second scheme.

We also gave a compact proof that the problem can have at most two planar solutions. Through a selected example with synthetic data, it was shown that both schemes may converge to either of the two solutions, depending on the initial condition. In practice, once a solution is obtained, the other can be computed using the equations given for the dual solution.

In the tests carried out, both algorithms have converged to a possible solution, and accurate solutions have been obtained in less than 40 iterations using the first scheme, and in less than 15 iterations in the second one. As mentioned earlier, the results have not been as satisfactory when the translational motion component is perpendicular to the planar surface. These cases required several hundred iterations of either scheme for accurate solutions.

Even though both schemes require approximately the same number of computations for convergence to a solution (second scheme converges faster but requires more computation at each iteration, as discussed earlier), the second one seems more appropriate for parallel implementation.

## 8. Acknowledgments

We would like to thank Alen Yuille, Mike Brady, Tomaso Poggio and Shimon Ullman for helpful comments and pointers to obscure references.

## Appendix

**Lemma 1:** $(\mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T)\mathbf{c} = \mathbf{c} \times (\mathbf{a} \times \mathbf{b})$.

**Proof:** $(\mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T)\mathbf{c} = \mathbf{a}(\mathbf{a} \cdot \mathbf{b}) - \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) = \mathbf{c} \times (\mathbf{a} \times \mathbf{b})$. ∎

**Lemma 2:** If $\mathbf{a}\mathbf{b}^T = \mathbf{c}\mathbf{d}^T$ , then either

(1) $|\mathbf{a}| = 0$ or $|\mathbf{b}| = 0$, and, $|\mathbf{c}| = 0$ or $|\mathbf{d}| = 0$, or

(2) $\mathbf{a} \parallel \mathbf{c}$ and $\mathbf{b} \parallel \mathbf{d}$.

**Proof:** $\mathbf{c}^T(\mathbf{a}\mathbf{b}^T)\mathbf{d} = \mathbf{c}^T(\mathbf{c}\mathbf{d}^T)\mathbf{d}$, or $(\mathbf{c} \cdot \mathbf{a})(\mathbf{b} \cdot \mathbf{d}) = (\mathbf{c} \cdot \mathbf{c})(\mathbf{d} \cdot \mathbf{d})$. Now if $|\mathbf{a}| = 0$ or $|\mathbf{b}| = 0$, then $|\mathbf{c}|^2 |\mathbf{d}|^2 = 0$, which implies that either $|\mathbf{c}| = 0$ or $|\mathbf{d}| = 0$. Conversely, if $|\mathbf{c}| = 0$ or $|\mathbf{d}| = 0$, then $|\mathbf{a}| = 0$ or $|\mathbf{b}| = 0$.

For the rest of the proof we assume that $|\mathbf{a}|$, $|\mathbf{b}|$, $|\mathbf{c}|$, and $|\mathbf{d}|$ are non-zero. Now $(\mathbf{a}\mathbf{b}^T)\mathbf{b} = (\mathbf{c}\mathbf{d}^T)\mathbf{b}$ so that $\mathbf{a}(\mathbf{b} \cdot \mathbf{b}) = \mathbf{c}(\mathbf{d} \cdot \mathbf{b})$. It follows that $\mathbf{a} \parallel \mathbf{c}$. Similarly, $\mathbf{a}^T(\mathbf{a}\mathbf{b}^T) = \mathbf{a}^T(\mathbf{c}\mathbf{d}^T)$ so that $(\mathbf{a} \cdot \mathbf{a})\mathbf{b}^T = (\mathbf{a} \cdot \mathbf{c})\mathbf{d}^T$. Therefore, $\mathbf{b} \parallel \mathbf{d}$. ∎

**Lemma 3:** If $\mathbf{M}$ and $\mathbf{N}$ are $(3 \times 3)$ skew-symmetric matrices and $\mathbf{M}^2 = \mathbf{N}^2$, then $\mathbf{M} = \pm\mathbf{N}$.

**Proof:** The result is immediate if $\mathbf{M}$ and $\mathbf{N}$ are zero. So from now on we assume that they are non-zero. If $\mathbf{M}^2 = \mathbf{N}^2$, then $\mathbf{M}^2\mathbf{x} = \mathbf{N}^2\mathbf{x}$ for all vectors $\mathbf{x}$. Using the isomorphism between $(3 \times 3)$ skew-symmetrical matrices and vectors we have

$$\mathbf{m} \times (\mathbf{m} \times \mathbf{x}) = \mathbf{n} \times (\mathbf{n} \times \mathbf{x}),$$

where the vectors $\mathbf{m}$ and $\mathbf{n}$ correspond to the matrices $\mathbf{M}$ and $\mathbf{N}$ respectively. So

$$(\mathbf{m} \cdot \mathbf{x})\mathbf{m} - (\mathbf{m} \cdot \mathbf{m})\mathbf{x} = (\mathbf{n} \cdot \mathbf{x})\mathbf{n} - (\mathbf{n} \cdot \mathbf{n})\mathbf{x},$$

or

$$[\mathbf{m}\mathbf{m}^T - (\mathbf{m} \cdot \mathbf{m})I]\mathbf{x} = [\mathbf{n}\mathbf{n}^T - (\mathbf{n} \cdot \mathbf{n})I]\mathbf{x}.$$

If this is to be true for all vectors $\mathbf{x}$, we must have, $[\mathbf{m}\mathbf{m}^T - (\mathbf{m} \cdot \mathbf{m})I] = [\mathbf{n}\mathbf{n}^T - (\mathbf{n} \cdot \mathbf{n})I]$. Taking the trace of both sides we get $\mathbf{m} \cdot \mathbf{m} = \mathbf{n} \cdot \mathbf{n}$. Consequently, $\mathbf{m}\mathbf{m}^T = \mathbf{n}\mathbf{n}^T$. Using Lemma 2, we see that $\mathbf{m} \parallel \mathbf{n}$. Taken together with $|\mathbf{m}| = |\mathbf{n}|$ this means that $\mathbf{m} = \pm\mathbf{n}$ and so $\mathbf{M} = \pm\mathbf{N}$. ∎

**Lemma 4:** If $\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T = \mathbf{c}\mathbf{d}^T + \mathbf{d}\mathbf{c}^T$ , then either

(1) $|\mathbf{a}| = 0$ or $|\mathbf{b}| = 0$, and, $|\mathbf{c}| = 0$ or $|\mathbf{d}| = 0$, or

(2) $\mathbf{a} \parallel \mathbf{c}$ and $\mathbf{b} \parallel \mathbf{d}$, or

(3) $\mathbf{a} \parallel \mathbf{d}$ and $\mathbf{b} \parallel \mathbf{c}$.

**Proof:** We first show that either $\mathbf{a}\mathbf{b}^T = \mathbf{c}\mathbf{d}^T$ or $\mathbf{a}\mathbf{b}^T = \mathbf{d}\mathbf{c}^T$. The proof then follows using lemma 2. We also need to show that $\mathbf{a} \cdot \mathbf{b} = \mathbf{c} \cdot \mathbf{d}$. This follows from the fact that $\text{Trace}(\mathbf{a}\mathbf{b}^T) = \mathbf{a} \cdot \mathbf{b}$ and $\text{Trace}(\mathbf{c}\mathbf{d}^T) = \mathbf{c} \cdot \mathbf{d}$. Now $(\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T)^2 = (\mathbf{c}\mathbf{d}^T + \mathbf{d}\mathbf{c}^T)^2$, or

$$(\mathbf{a} \cdot \mathbf{b})\mathbf{a}\mathbf{b}^T + (\mathbf{b} \cdot \mathbf{b})\mathbf{a}\mathbf{a}^T + (\mathbf{a} \cdot \mathbf{a})\mathbf{b}\mathbf{b}^T + (\mathbf{a} \cdot \mathbf{b})\mathbf{b}\mathbf{a}^T$$
$$= (\mathbf{c} \cdot \mathbf{d})\mathbf{c}\mathbf{d}^T + (\mathbf{d} \cdot \mathbf{d})\mathbf{c}\mathbf{c}^T + (\mathbf{c} \cdot \mathbf{c})\mathbf{d}\mathbf{d}^T + (\mathbf{c} \cdot \mathbf{d})\mathbf{d}\mathbf{c}^T.$$

Further, since $\mathbf{a}\cdot\mathbf{b} = \mathbf{c}\cdot\mathbf{d}$, we have $(\mathbf{a}\cdot\mathbf{b})(\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T) = (\mathbf{c}\cdot\mathbf{d})(\mathbf{c}\mathbf{d}^T + \mathbf{d}\mathbf{c}^T)$. Subtracting twice this amount from the previous equality we get

$$-(\mathbf{a}\cdot\mathbf{b})\mathbf{a}\mathbf{b}^T + (\mathbf{b}\cdot\mathbf{b})\mathbf{a}\mathbf{a}^T + (\mathbf{a}\cdot\mathbf{a})\mathbf{b}\mathbf{b}^T - (\mathbf{a}\cdot\mathbf{b})\mathbf{b}\mathbf{a}^T$$
$$= -(\mathbf{c}\cdot\mathbf{d})\mathbf{c}\mathbf{d}^T + (\mathbf{d}\cdot\mathbf{d})\mathbf{c}\mathbf{c}^T + (\mathbf{c}\cdot\mathbf{c})\mathbf{d}\mathbf{d}^T - (\mathbf{c}\cdot\mathbf{d})\mathbf{d}\mathbf{c}^T,$$

which reduces to $(\mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T)^2 = (\mathbf{c}\mathbf{d}^T - \mathbf{d}\mathbf{c}^T)^2$. Using the result of lemma 3 now, we get $(\mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T) = \pm(\mathbf{c}\mathbf{d}^T - \mathbf{d}\mathbf{c}^T)$. Adding $\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T = \mathbf{c}\mathbf{d}^T + \mathbf{d}\mathbf{c}^T$ to this equality we find that either $2\mathbf{a}\mathbf{b}^T = 2\mathbf{c}\mathbf{d}^T$ , or $2\mathbf{a}\mathbf{b}^T = 2\mathbf{d}\mathbf{c}^T$. The conclusion follows using Lemma 2. ∎

## References

[1] Adiv, G., "Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects," COINS TR 84-07, Computer and Information Science, University of Massachusetts, Amherst, MA, April 1984.

[2] Ballard, D.H., Kimball, O.A., "Rigid Body Motion from Depth and Optical Flow," TR 70, Computer Science Department, University of Rochester, Rochester, NY, 1981.

[3] Bruss, A.R., Horn, B.K.P., "Passive Navigation," *Computer Vision, Graphics, and Image Processing*, Vol. 21, pp. 3-20, 1983.

[4] Fang, J.Q., Huang, T.S., "Solving Three-Dimensional Small-Rotation Motion Equations: Uniqueness, Algorithms, and Numerical Results," *Computer Vision, Graphics, and Image Processing*, Vol. 26, pp. 183–206, 1984.

[5] Fang, J.Q., Huang, T.S., "Some Experiments on Estimating the 3-D Motion Parameters of a Rigid Body from Two Consecutive Image Frames," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 5, pp. 545-554, September 1984.

[6] Hay, C.J., "Optical Motion and Space Perception, an Extension of Gibson's Analysis," *Psychological Review*, Vol. 73, pp. 550–565, 1966.

[7] Hildreth, E.C., *The Measurement of Visual Motion*, MIT Press, 1983.

[8] Horn, B.K.P., Schunck, B.G., "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, pp. 185–203, 1981.

[9] Longuet-Higgins, H.C., "A Computer Algorithm for Reconstructing a Scene from Two Projections," *Nature*, Vol. 293, pg. 131-133, 1981.

[10] Longuet-Higgins, H.C., Prazdny, K., "The Interpretation of a Moving Retinal Image , Proceedings of the Royal Society of London, Series B 208, pp. 385-397, 1980.

[11] Maybank, S.J., "The Angular Velocity Associated with the Optical Flow Field Due to a Single Moving Rigid Plane," *Proceedings of the Sixth European Conference on Artificial Intelligence*, pp. 641–644, September 1984.

[12] Sugihara, K., Sugie, N., "Recovery of Rigid Structure from Orthographically Projected Optical Flow," TR 8304, Department of Information Science, Nagoya University, Nagoya, Japan, October 1983.

[13] Tsai, R.Y., Huang, T.S., Zhu, W.L., "Three-Dimensional Motion Parameters of a Rigid Planar Patch, ll: Singular Value Decomposition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, No. 4, August 1982.

[14] Tsai, R.Y., Huang, T.S., "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 1, January 1984.

[15] Waxman, A.M., Ullman, S., "Surface Structure and 3-D Motion from Image Flow: A Kinematic Analysis," CAR-TR-24, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD, October 1983.

[16] Waxman, A.M., Wohn, K., "Contour Evolution, Neighborhood Deformation and Global Image Flow: Planar Surfaces in Motion," CAR-TR-58, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD, April 1984.

[17] Wohn, K., Davis, L.S., Thrift, P., "Motion Estimation Based on Multiple Local Constraints and Nonlinear Smoothing," *Pattern Recognition*, Vol. 16, No. 6, pp. 563–570, 1983.